


Docket No.: S3-03P05196

CERTIFICATION

I, the below named translator, hereby declare that: my name and post office address are as stated below; that I am knowledgeable in the English and German languages, and that I believe that the attached text is a true and complete translation of PCT/DE2004/001341, filed with the German Patent Office on July 23, 2004.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Hollywood, Florida .


Rebekka Pierre

January 23, 2006

Lerner Greenberg Stemer LLP
P.O. Box 2480
Hollywood, FL 33022-2480
Tel.: (954) 925-1100
Fax.: (954) 925-1101

PCT/DE2004/001341
2003P05196WOUS

- 1 -

1 Description

2

3 Method for controlling a data interchange

4

5 The invention relates to a method for interchanging data
6 between a communication unit and a data source, in which a
7 runtime system comprising hardware components and software
8 components transmits data between the data source and a
9 communication unit and a processing sequence controls and/or
10 monitors the interchange of the data.

11

12 Such a method is already known from the accepted prior art.
13 Thus, by way of example, centralized control systems are
14 normally used for monitoring and controlling large-capacity
15 networks such as power supply mains, water supply lines and
16 rail systems. Larger blocks of flats may also be equipped with
17 centralized control systems for controlling air-conditioning
18 systems, elevators, lighting systems or the like. The parts
19 required for controlling such divided systems are therefore
20 normally likewise decentralized or in other words set up so
21 that they are distributed over a large area and are connected
22 to one another by means of a runtime system which has at least
23 one expedient communication network and programmable computer
24 units on which expedient runtime programs allow information to
25 be interchanged. Generally, hardware interfaces are used for
26 data interchange between the parts which deliver process
27 values, for example, on the one hand, and the locally installed
28 software components of the runtime system, on the other hand.
29 For the purpose of requesting these process values, a
30 communication unit is provided, such as an input computer
31 connected to the hardware interface via the communication
32 network. The

33

1 processing sequence is used to control the data interchange
2 between the parts and the communication unit.

3
4 Thus, the processing sequence checks the user name entered for
5 the purpose of logging into the runtime system, for example,
6 and the associated password for authorization to receive the
7 process values from the selected part. This allows sensitive
8 process values to be shielded from being known by particular
9 users. In addition, processing sequences are known which are
10 equipped with what is known as error analysis, which the
11 processing sequence uses to indicate the presence of
12 incompatibilities among parts which are used or among software
13 modules and possibly to demonstrate solutions for eliminating
14 such defects.

15
16 The previously known method has the attached drawback that the
17 processing sequence is monolithically embedded in a source code
18 of the runtime programs running during normal operation of the
19 centralized control system. This means that such processing
20 steps for controlling data interchange can be changed only by
21 altering the source code of the software components. Following
22 the change, the entire runtime programs therefore need to be
23 recompiled and installed on the hardware components.

24
25 It is an object of the invention to provide a method of the
26 type mentioned at the outset which can easily be altered or
27 extended without interrupting the runtime system.

28
29 The invention achieves this object in that the processing
30 sequence is made up of processing routines which each have a
31 standard input interface, with the processing routines being
32 called in succession and

1 the data in a called processing routine being supplied to the
2 input interface of a processing routine which is immediately
3 downstream of the latter, and in that the runtime system
4 manages a dynamic memory area and accesses said memory area in
5 order to stipulate the order in which the processing routines
6 are called.

7
8 In line with the invention, the control and monitoring of the
9 interchange of the data is of flexible design and can also be
10 altered as desired after the runtime system has been started
11 up. To this end, the data, for example a request which a user
12 has input for process values from the part, pass through
13 processing routines in order. The processing routines monitor
14 the requests, for example by storing them in access files, or
15 control them by adding further data, for example. In this case,
16 each processing routine has an input interface which is defined
17 by software and which is identical for all processing routines.
18 For the purpose of interchange, the data processed by the
19 respective processing routine are then supplied to the input
20 interface of the processing routine which is immediately
21 downstream. In other words, each processing routine is
22 compatible or interchangeable with the other processing
23 routines on account of its standard input interface. The
24 processing routines can therefore be called in any desired
25 order without the data interchange between the processing
26 routines causing error messages or more serious damage.

27 In line with the invention, the runtime system comprises
28 runtime programs and hardware components which are compiled
29 from computers, physical centralized control networks,
30 interfaces or the like. The physical centralized control
31 networks also

32

1 comprise wireless network connections. The runtime programs can
2 be distributed over the hardware components.

3
4 To be able to alter the order of the processing routines in
5 line with the respective requirements even during processing of
6 the software components in the runtime system, the software of
7 the runtime system manages a dynamic memory area whose memory
8 size can thus also be altered during operation of the runtime
9 system. To stipulate the order in which the processing routines
10 are called, the runtime system accesses processing data stored
11 in the memory area. The processing data may be a configuration
12 file, for example, in which the addresses of the desired
13 processing routines are listed line by line, with the runtime
14 system executing the lines in succession and in so doing
15 calling the processing routine listed in each line by means of
16 its address. The runtime system executes the lines of the
17 configuration data sequentially until the end of the
18 configuration file is indicated to the runtime system.

19
20 The dynamic management of the memory area allows any number of
21 lines to be provided and hence any number of processing
22 routines to be called. This may advantageously be used right at
23 the time at which the software components of the runtime system
24 are developed, by virtue of error diagnosis routines or in
25 other words error analysis routines being incorporated into the
26 processing sequence. Since the runtime programs are executed
27 largely without error, the number of error diagnosis routines
28 can be greatly reduced in order to increase the speed of data
29 interchange in this way or in other words to improve the
30 "performance" of the runtime system. This in no way requires
31 the

32

1 software components of the runtime system to be changed. By way
2 of example, the invention merely requires reparameterization to
3 be performed. This also applies to the search for errors
4 following implementation of the runtime system, said errors
5 being able to be restricted by the later addition of error
6 analysis routines to the processing sequence.

7
8 Both hardware components and software components are suitable
9 as a data source. Thus, the data source may be part of a
10 centralized control system, for example, with the runtime
11 system being connected to the part by means of expedient
12 interfaces. As a departure from this, the data source may also
13 be a software module, for example a software driver or else a
14 database containing information data corresponding to a
15 particular state or to the version of a system, however.

16
17 In line with the invention, the dynamically managed memory area
18 is a memory area in the "RAM store" of a computer.

19
20 Advantageously, the data are provided with a user identifier,
21 and at least one authorization routine checks the user
22 identifier for a match with entries in prescribed user lists
23 and terminates the forwarding of the data if it establishes
24 that there is no match between the user identifier and the user
25 lists. In this way, the user is shown only process values which
26 he is authorized to receive. Sensitive data can thus be
27 displayed on a user-specific basis. The user identifier does
28 not necessarily have to have an individualizing character
29 within the context of the invention. Thus, it is entirely
30 possible for the user identifier to have a role-specific
31 character such that the user is assigned as such to

32

1 a particular group or role. It is thus possible for the user to
2 be characterized as a developer or parameter-setter by the user
3 identifier, for example.

4
5 It is also expedient for the data to be provided with a
6 data-source-specific source data identifier, and for one or
7 more of the processing routines to control the interchange of
8 the data on the basis of the source data identifier. The source
9 data identifier, like the user identifier, is produced by
10 adding "metadata" to the data which are interchanged.

11
12 In line with one further development which is advantageous in
13 this regard, at least one processing routine is a buffer-store
14 routine in which buffer-store data are buffer-stored with a
15 respective buffer-store data identifier, and if the source data
16 identifier matches one of the buffer-store data identifier then
17 the buffer-store routine displays the buffer-store data
18 associated with the buffer-store data identifier and terminates
19 the interchange of the data. If the source data identifier is a
20 part identifier, for example, it is possible, by way of
21 example, for a request for particular process values from a
22 centralized control system to involve the processing routine
23 being prompted to buffer-store particular process values. By
24 way of example, the buffer-stored process values are process
25 values which change only slowly in comparison with the request
26 frequency or which do not change at all, or else parameters
27 which have been input by a third party with access
28 authorization. Upon a fresh request, the processing routine
29 (which is also called a buffer-store or caching routine, for
30 example) provides the requested process value without the
31 runtime system having accessed the relevant part. It has thus
32 become superfluous

33

1 for the runtime system to access the part, which speeds up the
2 method.

3
4 In this connection, any other display options are conceivable
5 which cannot be conclusively listed at this juncture. Thus, by
6 way of example, an "enrichment routine" can convert coded data
7 from the runtime system into user-comprehensible data. The
8 enrichment routine also adds additional control data or in
9 other words metadata to the data which are to be interchanged
10 between the communication unit and the data source in order to
11 control the display of data or the flow of data.

12
13 Advantageously, one of the processing routines is an error
14 analysis routine which checks the data for the presence of
15 errors. This may be any desired error analysis tool. Thus, the
16 data can be checked, by way of example, to determine whether
17 instead of a natural number or an integer there have been
18 letters or the like input. However, the error analysis routine
19 can also monitor the compatibility of protocols or hardware
20 components of the runtime system.

21
22 Advantageously, at least one processing routine is a monitoring
23 routine which stores the data and/or monitoring data derived
24 from the data in a monitoring file. This monitoring file
25 stores, by way of example, all access operations to the runtime
26 system in a month, which means that this makes it possible to
27 document who has accessed what data source, for example a part
28 in a centralized control system, at what time.

29

1 In line with one preferred exemplary embodiment of the
2 invention, the runtime system has a network server with a
3 server program and at least one client computer with a browser
4 program, and each browser program accesses the server program
5 via the Internet. In this exemplary embodiment of the
6 invention, the data interchange is made possible not just via
7 an externally terminated centralized control communication
8 network, for example, but rather via existing Internet
9 connections which have already been provided physically. It
10 goes without saying that the invention also allows, by way of
11 example, the communication network of the centralized control
12 system to be incorporated into a superordinate "Intranet",
13 which for its part can be connected to the Internet.

14
15 In line with one further development in this regard, at least
16 one processing routine is a tracing routine which checks the
17 path of the data in the runtime system and generates security
18 parameters on the basis of the check. On the basis of these
19 security parameters, it is now possible to control the display
20 or forwarding of the data, for example. If a user of the method
21 is connected to the runtime system via an "Intranet", for
22 example, fewer reservations regarding data sensitivity or
23 integrity are normally required, since access to the Intranet
24 by unauthorized parties is normally more difficult. In the case
25 of local applications, security reservations can be eliminated
26 almost completely, whereas only insensitive data or process
27 values are displayed during access via the Internet.

28
29 Expediently, a configuration file is loaded into the dynamic
30 memory area, the configuration file stipulating the structure
31 and the order of the processing routines. by way of example,
32 the configuration file is called

33

1 when the runtime system is initialized. In addition, however,
2 it is also possible for the user to initiate calling of the
3 configuration file by the runtime system. The configuration
4 file can also be called following implementation by the runtime
5 system without the user, for example at particular times.

6
7 Further expedient refinements and advantages of the invention
8 are the subject matter of the description below with reference
9 to the figure of the drawing, in which

10
11 Figure 1 shows a flowchart for schematically illustrating the
12 inventive method.

13
14 Figure 1 shows a flowchart to clarify the inventive method. It
15 schematically shows a centralized control system 1 which
16 comprises local protective devices 2 and 3 and also a central
17 control center 4 for controlling and monitoring the protective
18 devices 2, 3. The protective devices 2 and 3 are connected to
19 voltage transformers (not shown in the figure) whose primary
20 side is coupled to a system branch (likewise not shown) in a
21 power distribution mains. The secondary-side converter current,
22 which is proportional to the current in the system branch, is
23 sampled by a measurement detection unit in the protective
24 device 2 to obtain samples, and the samples are then digitized
25 to form digital current values. The protective devices 2 or 3
26 can trigger expedient switches or circuit breakers which
27 interrupt the flow of current in the system branch if, by way
28 of example, the digital current values exceed a threshold
29 value, for example in the case of a short circuit. In this
30 context, the protective devices 2 or 3 are arranged in the
31 immediate surroundings of the system branch, that is to say of
32 the primary conductor.

33

1 The control center 4 is provided for monitoring and controlling
2 the protective devices 2 or 3. To this end, it is connected to
3 the protective devices 2 and 3 via an expedient communication
4 network (not shown in the figure). To ensure secure data
5 interchange between the protective devices 2 or 3 and the
6 control center 4, a continually running runtime program 5 is
7 provided which is distributed over the hardware components in
8 the runtime system. In this case, the runtime program 5 uses
9 hardware drivers 6 to access hardware interfaces in the parts
10 2, 3, 4 of the centralized control system 1. Thus, by way of
11 example, digital current values from the protective device 2
12 are stored in a register in the protective device 2 and can be
13 supplied to the control center 4 via the communication network
14 which is not shown, the hardware drivers 6 undertaking the
15 addressing and control of the flow of data together with the
16 runtime program 5.

17
18 To be able to monitor the states of the parts 2, 3 or 4 of the
19 centralized control system 1 externally, that is to say from
20 locations which are not included in the communication network
21 of the centralized control system 1, communication units are
22 provided, such as a fixed-location standalone computer 7, a
23 laptop 8 or a "PDA", which are connected to the "Internet" via
24 a modem port, ISDN, DSL or a wireless local area network
25 connection. The runtime system comprises an Internet computer
26 on which a server program in the runtime program 5 runs. The
27 communication units use their browser programs 14 to access the
28 server program via the lines of the Internet. A user is
29 therefore able to request process values from the centralized
30 control system 1 and/or to control these process values via the
31 Internet.

32

1 To control the data interchange between the components 2, 3 and
2 4 of the centralized control system 1 and the communication
3 units 7, 8 and 9, a processing sequence 10 is provided which is
4 made up of successively running processing routines 11. To
5 allow smooth data interchange between the processing routines
6 11 in any order, their software includes a respective standard
7 output interface and also a respective standard input
8 interface, with the data to be controlled and monitored being
9 routed from the output interface of one of the processing
10 routines to the input interface of the downstream, subsequently
11 called processing routine.

12

13 To request the current value digitized by the protective device
14 2, a user with the PDA 9, for example, uses a wireless
15 "Bluetooth" connection to physically connect his PDA to the
16 Internet. The user then uses his PDA 9 to register on the
17 runtime program 5 by specifying his user name and his password.
18 Next, he selects the protective device 2, for example from a
19 protective device tree which is displayed to him, and the
20 process value which is required by the protective device 2. The
21 runtime program 5 takes the selection on the PDA 9 as a basis
22 for producing a part identifier as source data identifier which
23 is specific to the protective device 2. In other words, a part
24 address is produced on the basis of the selection by the user.
25 In addition, a register address for selecting the desired
26 current value is generated. The runtime program 5 also produces
27 control data, in this case as "Read signal", which is used to
28 notify the addressed hardware interface that the addressed
29 register needs to be read. Before the processing sequence 10 is
30 processed, the

31

1 data also have a user identifier added to them on the basis of
2 the user name.

3
4 Upon a request for the digital current value from the
5 protective device 2 to be shown, these request data are routed
6 through the processing sequence 10 in a request direction 12.
7 In the exemplary embodiment shown, the first processing routine
8 is a security routine 11a which accepts the data at its input
9 interface from the runtime program 5. The security routine 11a
10 establishes whether the user is authorized for the data
11 request. To this end, the security routine 11a compares the
12 user identifier of the request data with lists embedded in the
13 security routine and forwards the data to the output interface
14 of the authorization routine 11a only if the user identifier
15 matches an entry in this list.

16
17 From there, the data are routed to the input interface of a
18 buffer-store routine 11b. The buffer-store routine 11b checks
19 whether the requested process values are particular process
20 parameters which have been stipulated in software when the
21 buffer-store routine was created. Such process values are
22 process values which change only slowly in comparison with the
23 time interval between two successive requests or which do not
24 change at all, for example. If the buffer-store routine 11b
25 establishes that such a particular process parameter stored in
26 it from a previous request is being requested, it provides this
27 process parameter which has already been requested previously
28 and terminates the further request. Otherwise, it forwards the
29 data via its output interface directly to a user routine 11c.
30 The user routine 11c forwards the data in the request direction
31 12 to a "tracing routine" 11d without
32

1 processing the data, and the runtime program 5 adopts the
2 processed data again from the tracing routine. In the request
3 direction 12, the data are not processed by the tracing routine
4 11d either.

5
6 The runtime program 5 then uses the associated hardware
7 interface 6 to access the process values of the selected part 2
8 and adds a current value to the data as process value. Next,
9 the runtime program 5 transfers the data with the current value
10 to the input interface of the tracing routine 11d. The data are
11 now routed through the processing sequence 10 in direction 13.
12 The tracing routine 11d uses the request data to check the
13 location from which the user is accessing the runtime
14 program 5. If the user has registered on the runtime program 5
15 using a local area network which can be accessed only with
16 great difficulty externally, for example, the display options
17 of the runtime program are not limited by tracing routine 11d.
18 In the present example, the user of the PDA 9 has registered on
19 the runtime program 5 via the Internet, however, which means
20 that for reasons of security only restricted display of
21 information is intended. To this end, the tracing routine 11d
22 adds further security data to the requested current value and
23 to the rest of the request data, said security data producing a
24 particular display format for the runtime program 5.

25
26 Next, the data are routed to the user routine 11c, which adds
27 display parameters to the data, depending on the role of the
28 user. In the exemplary embodiment shown, the user is a
29 parameter-setter for whom highly specialized display data, for
30 example for detecting errors, might also be useful, whereas
31 they would be confusing to the normal user. The user routine
32 11c therefore adds such

33

1 display parameters to the request data as prompt the runtime
2 program 5 to display all the data.
3
4 From the user routine 11c, the data are then routed to the
5 buffer-store routine 11b. In the arrow direction 13 shown, the
6 buffer-store routine 11b and the security routine 11a do not
7 process the data. The security routine 11a finally passes them
8 to the runtime program 5, which displays the data on the PDA 9
9 in accordance with the processing parameters.
10